# SonarQube Integration with Jenkins on Ubuntu EC2 (2025 Guide)

## A) Install & Run SonarQube on Ubuntu EC2

Instance Requirements: t3.medium (min)   
OS: Ubuntu 22.04 LTS  
Open Ports: 9000 inbound to your IP (Security Group)  
  
1) System prep:  
sudo apt update && sudo apt -y upgrade  
sudo apt -y install unzip wget gnupg2 software-properties-common  
  
2) Install Java 17:  
sudo apt -y install openjdk-17-jdk  
java -version  
  
3) Install PostgreSQL:  
sudo apt -y install postgresql postgresql-contrib  
sudo systemctl enable --now postgresql  
  
4) Create DB & user:  
sudo -u postgres psql -c "CREATE USER sonar WITH ENCRYPTED PASSWORD 'StrongPass#123';"  
sudo -u postgres psql -c "CREATE DATABASE sonarqube OWNER sonar;"  
  
5) Linux limits & kernel params:  
echo 'vm.max\_map\_count=524288' | sudo tee -a /etc/sysctl.conf  
echo 'fs.file-max=131072' | sudo tee -a /etc/sysctl.conf  
sudo sysctl -p  
sudo tee -a /etc/security/limits.conf >/dev/null <<'EOF'  
sonarqube - nofile 131072  
sonarqube - nproc 8192  
EOF  
  
6) Create a dedicated user:  
sudo useradd -r -s /bin/false sonarqube  
  
7) Download & install SonarQube (LTS 9.9.5 example):  
cd /opt  
sudo wget  
sudo unzip sonarqube-9.9.5.87382.zip  
sudo mv sonarqube-9.9.5.87382 sonarqube  
sudo chown -R sonarqube:sonarqube /opt/sonarqube  
  
8) Configure database in SonarQube:  
sudo sed -i 's|#sonar.jdbc.username=.\*|sonar.jdbc.username=sonar|' /opt/sonarqube/conf/sonar.properties  
sudo sed -i 's|#sonar.jdbc.password=.\*|sonar.jdbc.password=StrongPass#123|' /opt/sonarqube/conf/sonar.properties  
sudo sed -i 's|#sonar.jdbc.url=jdbc:postgresql.\*|sonar.jdbc.url=jdbc:postgresql://localhost:5432/sonarqube|' /opt/sonarqube/conf/sonar.properties  
  
9) Systemd service:  
sudo tee /etc/systemd/system/sonarqube.service >/dev/null <<'EOF'  
[Unit]  
Description=SonarQube service  
After=network.target  
  
[Service]  
Type=simple  
User=sonarqube  
Group=sonarqube  
ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start  
ExecStop=/opt/sonarqube/bin/linux-x86-64/sonar.sh stop  
RemainAfterExit=yes  
LimitNOFILE=131072  
LimitNPROC=8192  
Restart=on-failure  
  
[Install]  
WantedBy=multi-user.target  
EOF  
  
sudo systemctl daemon-reload  
sudo systemctl start sonarqube  
sudo systemctl enable sonarqube  
sudo systemctl status sonarqube  
  
10) Access UI & create a token:  
http://<EC2-Public-IP>:9000  
Login: admin / admin → change password  
Generate token from My Account → Security

## B) Connect Jenkins to SonarQube

1) Install Jenkins Plugins:  
- SonarQube Scanner for Jenkins  
- Quality Gates  
  
2) Add SonarQube server in Jenkins:  
Manage Jenkins → System → SonarQube servers  
Name: SonarQube  
Server URL: http://<EC2-Public-IP>:9000  
Authentication Token: add as Secret Text credential (ID: sonar-token)  
  
3) Add Webhook in SonarQube:  
Administration → Configuration → Webhooks → Create  
Name: Jenkins  
URL: http://<jenkins-url>/sonarqube-webhook/  
  
4) Ensure Maven is available to Jenkins agent

## C) Updated Jenkinsfile with SonarQube Integration

pipeline {  
 agent any  
  
 environment {  
 SONAR\_TOKEN = credentials('sonar-token')  
 }  
  
 stages {  
 stage("Checkout Code") {  
 steps {  
 git 'https://github.com/akshu20791/addressbook-cicd-project'  
 echo "Cloning code"  
 }  
 }  
  
 stage("Compile") {  
 steps {  
 sh 'mvn -B -e compile'  
 }  
 }  
  
 stage("Test") {  
 steps {  
 sh 'mvn -B test'  
 }  
 }  
  
 stage("SonarQube Analysis") {  
 steps {  
 withSonarQubeEnv('SonarQube') {  
 sh '''  
 mvn -B -DskipTests sonar:sonar -Dsonar.projectKey=addressbook-cicd-project -Dsonar.host.url=$SONAR\_HOST\_URL -Dsonar.login=$SONAR\_TOKEN  
 '''  
 }  
 }  
 }  
  
 stage("Quality Gate") {  
 steps {  
 timeout(time: 10, unit: 'MINUTES') {  
 waitForQualityGate abortPipeline: true  
 }  
 }  
 }  
  
 stage("Package") {  
 steps {  
 sh 'mvn -B package'  
 }  
 }  
  
 stage("Deploy") {  
 steps {  
 sh 'sudo cp /var/lib/jenkins/workspace/project-pipeline/target/addressbook.war /home/ubuntu/apache-tomcat-9.0.108/webapps'  
 }  
 }  
 }  
}